УДК 004.056:004.65

**С.О. Спасітєлєва, В.Л. Бурячок**

# КОМПЛЕКСНИЙ ЗАХИСТ ГЕТЕРОГЕННИХ КОРПОРАТИВНИХ СХОВИЩ ДАНИХ

Запропоновано підхід щодо комплексного захисту сучасних корпоративних баз та сховищ даних, які побудовані за принципом багатоаспектної персистентності з використанням різних технологій зберігання та аналізу даних. На підставі проведеного аналізу загроз та засобів захисту даних для реляційних та NoSQL систем управління базами даних, визначені проблеми захисту даних для гетерогенного сховища даних та шляхи їх подолання.

***Ключові слова:*** інформаційна безпека, сховище даних, бази даних, РСУБД, NoSQL DBMS, великі дані, захист даних.

## Вступ і постановка завдання

В сучасних умовах будь-яка діяльність пов’язана з оперуванням великим обсягом структурованих та неструктурованих даних [1], які використовуються різними групами користувачів. Згідно з прогнозом International Data Corporation, до 2020 року таких даних в світі буде генеруватися понад 44 трильйонів гігабайт щорічно [2]. Для того, щоб ефективно справлятися з обробкою таких великих масивів інформації підприємства вже нині мають докладати певних зусиль для зміни своєї IT-інфраструктури, що, як результат, дозволить позитивно впливати на їх бізнесові проекти.

Нажаль традиційні способи та підходи оперування даними, що засновані на рішеннях класу бізнесової аналітики та реляційних системах управління базами даних є неефективними. Це пояснюється тим, що нині – в еру багатоваріантної персистентності (Polyglot Persistence), для задоволення різних потреб розробникам баз даних (БД) приходиться маніпулювати різними технологіями роботи з ними – від зберігання до управління [3]. Прикладом є технології NoSQL та MapReduce, поява яких активізувала напрямок розробки БД і окреслила нові перспективи їх розвитку для проведення розподіленої паралельної обробки великих масивів даних з використанням кластерів звичайних недорогих комп'ютерів.

Разом з тим надзвичайно небезпечною для підприємств і організацій останнім часом стає поява нових технологій атак на їх найуразливіші активи. Так, за висновками компанії Verizon [4], кількість витоків даних у світі неухильно зростає. Аналітичним центром InfoWatch [5] лише за I півріччя 2016 року зареєстровано 840 випадків витоку конфіденційної інформації, що на 16% більше, ніж за аналогічний період 2015 року. При цьому, власне, понад 30% з них припадають на зовнішніх порушників і більше 60% – виконано за участю співробітників організації. Причина, за якою БД так часто піддаються злому полягає в тому, що вони мають ключове значення для будь-якої організації, оскільки містять конфіденційну ділову інформацію [6].

Все це диктує потребу в нових більш надійних засобах безпеки БД, які здатні задовольнити вимоги до їх продуктивності та масштабованості.

**Аналіз останніх досліджень і публікацій.** Питання безпеки БД висвітлено в багатьох публікаціях закордонних і вітчизняних авторів. У класичних роботах розглянуто підходи до:

забезпечення конфіденційності, цілісності і доступності реляційних СУБД (РСУБД, RDBMS), визначення та попередження типових атак [7,8] для реляційних БД;

реалізації основних моделей доступу (дискреційного, мандатного, рольового) до реляційних серверів для різних розробників РСУБД [9];

забезпечення аудиту, шифрування даних, а також використання вбудованих механізмів таких, як представлення, обмеження, тригери, збережені процедури для конкретних РСУБД [10,11] тощо.

Серед закордонних авторів, які висвітлюють сучасні напрямки досліджень з Big Data, NoSQL та MapReduse можна відмітити роботи Фаулера М., Садаладжа П., Марца Н. та Уоррена Дж.. Приділяючи велику увагу власне технологіям сховищ даних, вони нажаль

практично не займаються забезпеченням їх захисту [1, 3]. Слід відмітити й фактичну відсутність досліджень щодо забезпечення безпеки сучасних сховищ даних, побудованих з використанням різних технологій баз даних (SQL, NoSQL) тощо.

**Актуальність та мета статті.** Все вище викладене фактично дає можливість стверджувати, що в контексті нових загроз та тенденцій розвитку інформаційної безпеки (ІБ) та збільшення ролі і розмаїття технологій створення сховищ даних, проблеми захисту даних стають нині особливо актуальними. В сучасних умовах виникає необхідність комплексного розгляду та систематизації питань безпеки для гетерогенних сховищ даних. Відповідно, метою статті є розгляд переваг і недоліків традиційних РСУБД, засобів бізнес аналітики та технології NoSQL, а також формування комплексного підходу щодо захисту сучасних корпоративних сховищ даних, які об'єднують зазначені технології.

**Виклад основного матеріалу досліджень**

В основу роботи реляційних та нереляційних БД покладено різні моделі даних, які визначають ключові підходи до всіх аспектів їх функціонування, в тому числі і до безпеки. Відповідно всі сучасні технології зберігання даних залежно від моделі даних можна розділити на декілька основних груп (рис. 1).

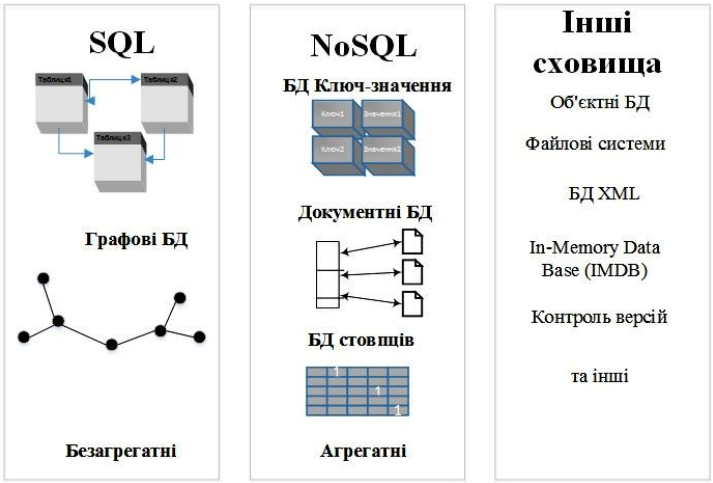


Рис. 1. Технології зберігання даних.

Незважаючи на відмінності між різними реляційними БД, їх основний механізм залишається одним і тим же, тобто вони використовують різні діалекти стандарту мови запитів SQL, а транзакції обробляються ними практично однаково. Реляційні БД призначені для ефективної роботи з структурованими даними і широко використовуються організаціями для управління операційними даними. Oracle Database, IBM DB2 і Microsoft SQL Server складають 90% ринку реляційних клієнт-серверних СУБД.

Програмні засоби бізнесової аналітики (Business Intelligence, BI) забезпечують функції доступу та аналізу інформації, яка міститься в спеціалізованому сховищі даних (Data Warehouse, скорочено DWH), а також забезпечують прийняття правильних і обґрунтованих управлінських рішень. До таких програмних продуктів відносять засоби побудови сховищ даних (Data warehousing), системи оперативної аналітичної обробки (OLAP), інформаційно-аналітичні системи (Enterprise Information Systems, EIS), засоби інтелектуального аналізу даних (Data Mining), інструменти для виконання запитів і побудови звітів (Query and Reporting Tools). Починаючи з 2013 року головним напрямком бізнес аналітики стає обробка неструктурованих даних за допомогою сховищ Hadoop, Teradata для організації розподіленої обробки великих обсягів даних з використанням парадигми MapReduce, при якій завдання ділиться на багато дрібніших відособлених фрагментів, кожен з яких може бути запущений на окремому вузлі кластера. У 2017 році основою стратегії в області великих даних і аналітики стане єдина

платформа даних, на якій працюють засоби управління інформацією, аналізу та пошуку для даних з web ресурсів, мультимедійних файлів – відео, аудіо та зображень.

Перевагою технології NoSQL є можливість обробки величезних обсягів неструктурованих даних на кластерах, лінійна масштабованість, підвищена відмовостійкість, нереляційність, можливість ефективного використання в додатках real-time web. Саме це робить технологію NoSQL достатньо привабливою для багатьох підприємств, але одночасно з цим висуває до неї значні вимоги з точки зору безпеки даних. Прикладами систем, які використовують зазначену технологію є: MongoDB, Oracle NoSQL Database, Couchbase, Cassandra, HBase, Redis, Riak, MemcacheDB, MUMPS, Vertica, AllegroGraph. Більшість сучасних NoSQL баз даних використовують агрегатну модель даних. Перевага агрегатної орієнтації полягає в тому, що вона дуже полегшує роботу на кластерах, яка була основною причиною появи технології NoSQL. При роботі на кластерах необхідно мінімізувати кількість вузлів, які необхідно опитати для збирання даних.

Поєднання цих технологій сприяє формуванню інтегрованого гетерогенного сховища корпоративних даних, яке може складатися з таких програмних засобів:

RDBMS різних виробників для збереження та обробки трансакційних структурованих даних;

NoSQL DBMS для збереження неструктурованих великих даних;

сховищ даних бізнес аналітики DWH або Hadoop.

Архітектура такого середовища представлена на рисунку 2.

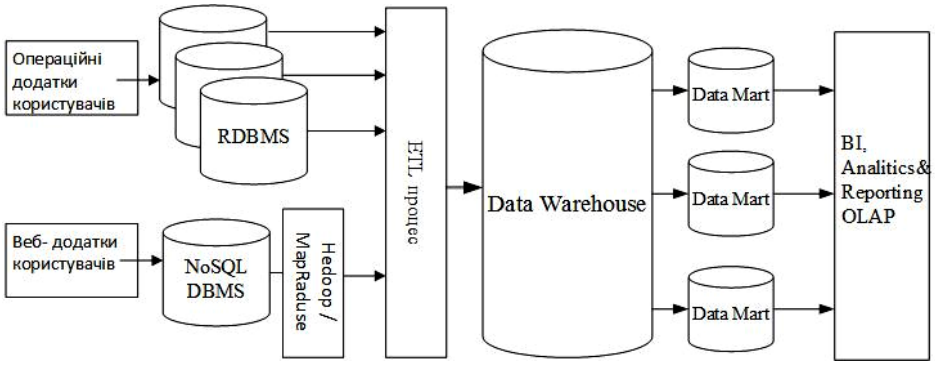


Рис. 2. Архітектура гетерогенного корпоративного сховища даних.

**Порівняльний аналіз засобів безпеки SQL та NoSQL баз даних.** Для того щоб зберігання даних було безпечним, БД має забезпечувати конфіденційність, цілісність і доступність (CIA). Корпоративні РСУБД забезпечують функції CIA за допомогою таких інтегрованих функцій безпеки, як:

шифрування даних;

управління доступом на основі ролей;

управління доступом до рядків та полів;

управління доступом до збережених процедур на рівні користувача.

Сучасні реляційні сервери мають достатньо розвинену систему безпеки, яка включає основні і додаткові модулі й містить засоби гранулювання доступу до рівня запису та маскування даних.

NoSQL рішення з'явилися на ринку недавно і ще не встигли пройти «шлях помилок і вразливостей», характерний для їх більш зрілих реляційних аналогів [12]. У базах NoSQL, як і у РСУБД де дані обслуговуються і видаються користувачам за запитами, також існує механізм запитів. Ідея полягає в тому, що додаток володіє даними і обслуговує їх за допомогою сервісів. В рамках такого підходу відповідальність за безпеку перекладається в основному на додаток.

БД РСУБД також мають набір властивостей ACID (atomicity, consistency, isolation, durability), які гарантують надійну обробку транзакцій БД, наприклад, реплікація і запис транзакцій в журнал забезпечують надійність і цілісність. Реляційні БД дозволяють

маніпулювати будь-якою комбінацією рядків з будь-якої таблиці в рамках однієї ACID. Багато рядків з різних таблиць оновлюються в рамках однієї операції. Ця операція завершується або повним успіхом, або повною невдачею, причому паралельні операції ізолюються одна від одної так, що вони не можуть виконувати часткові модифікації. Ці функції збільшують час, необхідний для доступу до великих обсягів даних, тому вони не реалізуються в БД типу NoSQL. Реалізація переваг швидкого і легкого доступу до даних негативно відбивається на безпеці NoSQL бази даних.

На транзакції суттєво впливає наявність агрегатів. У агрегатно-орієнтованих БД відсутній контроль цілісності даних, побудованих на поняттях ACID транзакції, зовнішніх ключах. Для забезпечення узгодженості даних вони підтримують атомарні маніпуляції з окремими агрегатами по черзі. Це означає, що для роботи з множиною агрегатів, потрібно управляти ними з коду програми. Графові та інші безагрегатні БД зазвичай використовують транзакції ACID.

Для того щоб забезпечувати швидкий доступ до даних, NoSQL бази даних створюються з невеликою кількістю функцій безпеки. Вони мають так званий набір властивостей BASE (basically available, soft state, eventually consistent). Замість того щоб підтримувати вимогу послідовності після кожної транзакції, БД має просто згодом досягати послідовного стану. Оскільки транзакції записуються в БД не відразу, є можливість взаємного перетину одночасних транзакцій. При цьому користувачі одночасно можуть бачити не однакові дані, тобто бази даних NoSQL не можуть використовуватися для обробки фінансових транзакцій. NoSQL сховища широко використовуються для збору та аналізу даних, які генеруються веб-додатками, соціальними мережами, датчиками, інтелектуальними лічильниками, телекомунікаційними сервісами тощо.

У БД NoSQL також відсутні функції конфіденційності і цілісності даних. Так як у таких БД немає логічної структури, права доступу до таблиці, колонки або рядка не можна розділяти. Це може призводити до появи декількох копій одних і тих же даних, ускладнювати підтримку послідовності даних, зокрема тому, що зміни в декількох таблицях не можуть об'єднуватися в одну транзакцію, у якій логічний блок операцій вставки, оновлення або видалення виконується в цілому.

Оскільки існує багато різних реалізацій NoSQL, відсутність стандартів також підвищує складність підтримки БД. Конфіденційність і цілісність даних повинні повністю забезпечуватися додатком, який звертається до даних NoSQL. Розробники додатків не відрізняються уважністю до реалізації функцій безпеки, і новий програмний код зазвичай означає нові помилки. Будь-які запити, що направляються в базу даних NoSQL, повинні перенаправлятися, фільтруватися і підтверджуватися, в той час як сама БД повинна завжди знаходиться в захищеному середовищі.

У базах даних NoSQL основу системи безпеки складають технології шифрування і токенізації [13]. Коли безпека прив'язана безпосередньо до даних, інші елементи системи захисту можуть давати збої, але при цьому конфіденційні дані не будуть скомпрометовані. Використовується кілька різних способів впровадження шифрування і токенізації для моментального захисту даних на рівні окремих додатків. Найчастіше організації застосовують шифрування або на рівні файлової системи, або на рівні додатку.

Шифрування на рівні файлової системи являє собою виключно гнучкий спосіб захисту важливих даних у базах даних NoSQL. Подібний підхід дозволяє організаціям групувати і розділяти свої дані за рівнем важливості, забезпечуючи захист тільки необхідних даних. При подібному захисті даних засоби шифрування можуть забезпечувати додаткові механізми контролю, обмежуючи доступ до даних для певних користувачів, груп за будь-якими параметрам. Шифрування на рівні файлової системи являє собою виключно гнучкий і непомітний спосіб захисту файлів на шляху до місця зберігання, при цьому такий підхід дозволяє захищати дані різних типів – від зображень і логотипів до баз даних і поштових архівів. Шифрування даних в додатку може стати ідеальним способом для захисту окремих полів БД без зміни самої архітектури БД. При здійсненні шифрування на цьому рівні

інформація за жодних умов не зберігається і не передається в незашифрованому вигляді, що дозволяє організації значно зменшити область потенційної атаки. Захищаючи дані в цій точці, організації можуть замість шифрування скористатися токенізацією, щоб «затуманити» (obfuscate) дані перед передачею їх у сховище.

Токенізація ідеально підходить для захисту конфіденційних даних, таких як, номери кредитних карт, номери соціального страхування, а також будь-яких інших даних, які зловмисники регулярно викрадають з метою використання або продажу на "чорному ринку". Сервери токенізації можуть бути розгорнуті всередині компанії або використовуватися "як сервіс". Якщо сервери токенізації використовуються для різних подій в декількох різних системах, то вони можуть видавати різні токени для одних і тих же даних. Сервери токенізації виконують такі функції: створення і зберігання токенів, шифрування, перевірка користувачів і повернення конфіденційних даних авторизованим додаткам.

При створенні гібридних сховищ даних необхідно ретельно розглянути вимоги до безпеки, конфіденційності та цілісності перед вибором технологій збереження та обробка даних. Різні масштаби і види збережених даних вимагають різних підходів до безпеки. Відсутність в деяких NoSQL функцій безпеки, а саме, підтримки автентифікації або авторизації, означає, що конфіденційні дані вимагають додаткових засобів захисту. Останнім часом розробники NoSQL систем стали впроваджувати функції безпеки, які є притаманними системам RDBMS. Наприклад, Oracle впровадила операційний контроль над даними, записуваними в один вузол. Oracle NoSQL Database має тісну інтеграцію з РСУБД Oracle. Користувачі РСУБД Oracle можуть переглядати записи в СУБД Oracle NoSQL і виконувати запити безпосередньо з середовища SQL через зовнішні таблиці, що забезпечує негайну доступність даних NoSQL і їх готовність до інтегрованого аналізу. У Oracle NoSQL Database підтримується незалежна від операційної системи парольна автентифікація в масштабі кластера і інтеграція сховища ключів Oracle Wallet, що покращує захист від несанкціонованого доступу до конфіденційних даних. Cassandra підтримує ведення журналу і автоматичну реплікацію транзакцій, а система MongoDB підтримує реплікацію головних і підпорядкованих пристроїв.

**Загрози безпеці сховищ даних.** Список основних вразливостей сховищ даних актуальний не тільки для реляційних БД, але і для рішень NoSQL. За підсумками 2015 року визначено десять найважливіших загроз безпеки баз та сховищ даних [14]: надмірні і невикористовувані привілеї, призначені для користувача; зловживання привілеями; input-ін'єкції (ін'єкції в полі вводу); хакерські програми; недостатні заходи з аудиту даних; незахищеність носіїв інформації; експлуатація вразливих та неправильно сконфігурованих БД; некерована конфіденційна інформація; відмова в обслуговуванні (DoS); брак знань і досвіду в сфері ІБ.

Можна визначити два основних способи злому БД за допомогою ін'єкцій коду:

1) SQL-ін'єкції, застосовувані для злому реляційних СУБД.

2) NoSQL-ін'єкції, які використовуються для злому платформ Big Data. NoSQL-ін'єкції зазвичай є впровадженням недозволеного або шкідливого коду в поля введення веб-додатків, компонент MapReduce, Hive. Реалізуються шляхом [15]:

використання регулярних виразів в параметрах запиту;

отримання доступу до даних через спеціальний інтерфейс, який підтримується NoSQL (наприклад, MongoDB використовує як мову запитів BSON, eXist застосовує XQuery, а Sonic GraphDB – GraphQL, тому для MongoDB можна реалізувати JSON-ін'єкції і т. д.);

маніпулювання з REST-інтерфейсом (Representational state transfer) і підробки міжсайтових запитів (CSRF);

виконання скриптів на сервері, на якому встановлена NoSQL (наприклад, MongoDB дозволяє запускати JavaScript-код, тому можливе виконання JavaScript-ін'єкції).

В обох випадках успішно проведена ін’єкція може дати необмежений доступ до вмісту БД. Незважаючи на те, що технічно NoSQL є невразливими для SQL-ін'єкцій тому, що в них не застосовується мова SQL, вони відкриті для атак, які організовані за аналогічними принципами.

Одним із джерел вразливостей даних є недосконала реалізація розмежування доступу до них. Існує ймовірність використання користувачами (додатками) своїх легітимних прав доступу в протиправних цілях або зловживання привілеями, обсяги яких перевищують необхідні для виконання посадових (функціональних) обов'язків. Як і будь-який додаток, NoSQL DBMS та RDBMS можуть піддаватися атакам переповнення буфера або мати вразливу схему автентифікації. Атакувати рівень СУБД досить складно, тому що користувачі і компанія-розробник намагаються виправляти помилки по мірі їх виявлення. Крім того, більшість програмних продуктів NoSQL поставляються з відкритим кодом, а значить, його можна проаналізувати. Більшість СУБД мають велику кількість різних бібліотек (клієнтське API) для доступу до даних. Ймовірність виявити вразливість в клієнтській бібліотеці є значно більшою, ніж безпосередньо в самій СУБД. При цьому можна не тільки знайти вразливість, яка відкриє доступ до всіх програм, побудованих на основі цього API, але зрозуміти, як саме відбувається діалог між клієнтським кодом і сервером баз даних та який використовується протокол. На рівні додатку для пошуку вразливостей зломщики шукають ті місця, де програміст забув перевірити вхідні дані, і намагаються їх використовувати. У такому випадку для боротьби використовується той же підхід, що і при пошуку SQL-ін'єкцій, тобто аналізується код і повідомлення про помилки з використанням мов додатків, наприклад, JSON, JavaScript тощо.

Корпоративна система повинна включати в себе засоби для автоматичної реєстрації транзакцій БД, в тому числі протоколювання операцій з конфіденційною інформацією. Відмова від збору детальних даних аудиту веде до виникнення серйозних загроз на різних рівнях.

Багато організацій використовують вбудовані в СУБД засоби аудиту, покладаються на вузькоспеціалізовані рішення або проводять аудит в ручному режимі. Однак можливості таких інструментів обмежені – вони не дозволяють проводити повноцінний аудит, виявляти спроби злому і проводити розслідування. Більш того, вбудовані в СУБД засоби часто здійснюють зайве навантаження на процесор і жорсткий диск сервера, тому в багатьох випадках функція аудиту просто відключається. Коли для роботи з БД використовуються корпоративні web-додатки, складно пов'язати конкретні операції щодо БД з конкретними співробітниками. Більшість вбудованих інструментів аудиту не здатні визначити кінцевого користувача, оскільки асоціюють активність БД з обліковими записами клієнтських додатків. Відсутність зв'язку з користувачем, які вчинили ту чи іншу операцію, перешкоджає веденню звітності, можливості спостереження і проведення розслідувань. До того ж, користувачі, що володіють правами адміністратора БД (легітимними або отриманими в результаті злому), можуть відключити вбудований аудит, щоб приховати свою активність. Саме тому необхідно розмежовувати функції управління аудитом і адміністрування БД і серверної платформи, щоб домогтися чіткого поділу зон відповідальності.

Більшість вбудованих рішень працюють лише на одній, призначеної для них, платформі. Це значно ускладнює впровадження однорідного, масштабованого механізму аудиту в організаціях, які використовують СУБД різного типу.

У разі використання різнорідних, мультиплатформних систем, виникає необхідність у шифруванні, розшифрування і повторному шифруванні даних при їх передачі та обробці, що створює додаткові вразливості, які можуть бути використані зловмисниками. Причому, чим більше ключів використовується в системі, тим більше можливостей для її атаки.

Розвиток засобів внутрішньої безпеки не встигає за зростанням обсягів даних, при цьому багато організацій недостатньо оснащені і підготовлені для протидії загрозам. Причинами цього є як недостатня інформованість або увага адміністраторів СУБД і прикладних програмістів, так і відсутність вбудованих засобів контролю відомих вразливостей в більшості СУБД. Хорошим рішенням були б автоматизація і перенесення контролю таких загроз на рівень сервера, однак різноманіття мов і мовних діалектів не дозволяє це зробити [12].

**Програмні рішення для комплексного захисту розподілених гібридних сховищ даних.** Використання різних технологій збереження та аналізу даних у сучасних корпоративних сховищах висуває нові вимоги з інформаційної сумісності та комплексного захисту інтегрованих даних. Отже, виникає потреба у створенні мультиплатформних засобів безпеки, які застосовуються до різних типів СУБД. Головні проблеми виникають при інтеграції середовища

NoSQL DBMS та Hadoop з традиційним реляційним середовищем RDBMS та DWH. Розглянемо п’ять категорій програмних рішень для забезпечення безпеки корпоративних сховищ [14].

*Засоби для управління правами доступу* визначають зайві права доступу до конфіденційної інформації, допомагають запобігти витоку даних і підвищити загальну безпеку БД за допомогою засобів контролю над привілейованими користувачами, конфігураціями і розподілом обов'язків. Необхідно комплексне рішення, що забезпечує додаткове розмежування повноважень всередині різних БД з реалізацією обмеження доступу до даних, а також обмеження доступу та контроль виконання команд в залежності від часу, IP-адреси, операції і так далі. При цьому виникає необхідність у сумісному використанні різних засобів управління доступом. Такими засобами є: рольове управління обліковими записами і правами доступу користувачів; управління доступом до даних на основі міток класифікації для задоволення вимог державного сектора щодо багаторівневої безпеки; одноразова автентифікація в розподіленому гетерогенному середовищі. Також необхідно використовувати засоби детального контролю доступу до даних – Virtual Private Database.

*Засоби моніторингу і блокування* захищають БД від злому, несанкціонованого доступу і викрадення інформації. Необхідно мати засоби контролю для створення безпечного середовища БД, захистити від внутрішніх і зовнішніх атак і запобігти несанкціонованим змінам даних. Засоби моніторингу активності користувачів і клієнтських додатків Database Firewall дозволяють повністю блокувати потенційно небезпечні операції або виконувати контрольну перевірку стандартних параметрів, таких як IP-адреси, метод автентифікації та ім'я програми. Засоби контролю безпеки дозволяють здійснювати моніторинг конфігурації конфіденційних баз даних, пошук і каталогізацію конфіденційних даних за допомогою сканування баз даних відповідно до різних параметрів безпеки, включно з перевіркою стандартних паролів, статусів і профілів облікових записів. Сучасні засоби постачаються з великою кількістю готових методик перевірки політик існуючих баз, які потрібно розширити і застосувати до різних типів даних.

*Засоби виявлення та оцінки* виявляють уразливості БД і місцезнаходження критично важливих даних. Аналіз привілеїв допомагає підвищити безпеку додатків шляхом визначення фактично використовуваних привілеїв під час виконання. Привілеї, які визначені як невикористовувані, можна розглянути на предмет можливого їх скасування, що допоможе скоротити спектр потенційних атак і реалізувати модель мінімальних привілеїв. Необхідно визначати області безпеки для запобігання вільного доступу до даних з боку привілейованих користувачів.

Для консолідації і аналізу подій ІБ можна використовувати системи управління подіями і інцидентами ІБ (SIEM, Security Information and Event Management). Технологія SIEM забезпечує аналіз в реальному часі подій безпеки отримуваних від різних джерел, управління внутрішніми і зовнішніми ризиками інформаційної безпеки організації, контроль відповідності виконання політик безпеки, побудову детальних звітів. SIEM системи виявляють такі події як: підбір паролів облікових записів, аномальна активність систем, зміна важливих параметрів інфраструктури.

*Засоби аудиту* допомагають підтвердити відповідність системи галузевим стандартам безпеки. Ці засоби дозволяють здійснювати перевірку шляхом консолідації і моніторингу даних аудиту з різних баз даних. Отримані дані аудиту надають можливість для комплексного огляду діяльності баз даних з повним контекстом виконання команд. Дані аудиту мають зберігатися на окремому сервері. Сучасні сховища даних мають використовувати функції умовного аудиту на базі політик. Політики можуть містити установки аудиту, а умови дозволяють прискорити процес аудиту на основі параметрів, пов'язаних з сесією бази даних. Наприклад, можна визначити політику аудиту, яка перевіряє всі дії за межами зазначеної IP-адреси і імені користувача. З'єднання поза політикою можуть перевірятися повністю, в той час як для інших з'єднань не будуть створюватися ніякі дані аудиту, що дозволить зробити аудит вибірковим і високоефективним. Система аудиту має проводити безперервний аудит даних, інтегруватися з SIEM-системами і передавати їм дані для аналізу.

Для автоматизації аудиту краще використовувати платформи DAP (Data Analysis and Probability). DAP-рішення підтримують різні СУБД від різних постачальників – це дозволяє використовувати єдині стандарти і централізовані операції з аудиту в масштабованих і розподілених гетерогенних середовищах БД.

*Засоби захисту даних* підтримують цілісність і конфіденційність даних. Одним із ефективних підходів до захисту даних є шифрування даних при їх зберіганні та редагування (маскування) конфіденційних даних в режимі реального часу на основі контексту сеансу бази даних тоді, коли вони передаються з бази даних. Необхідно забезпечити комплексне шифрування протягом всього життєвого циклу даних, включаючи резервне копіювання і експорт, інтеграцію з апаратними модулями безпеки або корпоративними рішеннями з управління. Прозоре шифрування даних (TDE) допомагає запобігти несанкціонованому доступу до експортованих баз даних. Необхідно забезпечити централізоване управління ключами шифрування, сховищами ключів і файлами облікових даних для всього підприємства.

**Висновок**

Інтеграція нових технологій збереження, обробки та аналізу великих даних в існуючі реляційні сховища даних є актуальною задачею сьогодення і висуває нові вимоги з інформаційної сумісності та комплексного захисту інтегрованих даних. Проведений аналіз SQL та NoSQL сховищ інформації показав, що різні масштаби та види збережених даних вимагають різних підходів до безпеки, використання різних моделей даних та мов доступу до них ускладнює розробку єдиного механізму захисту даних рівня сервера. Стандартизація підходів, моделей та мов роботи з даними різних форматів дозволить створити мультиплатформні засоби забезпечення безпеки для різних типів СУБД.

У статті проаналізовано та сформовано вимоги та рекомендації щодо комплексного підходу забезпечення безпеки гібридних сховищ даних, а також визначено головні загрози та програмні засоби для побудови розвинутих механізмів безпеки таких сховищ.

**Література**

1. Marz N., Warren J. Big Data: Principles and best practices of scalable realtime data systems», ISBN 9781617290343, April 2015. – 328pp.

2. Звіт Big Data (мировой рынок). [Електронний ресурс] // – Режим доступу: http://www.tadviser.ru/index.php/Big_Data (13.01.2017).

3. Фаулер Мартин NoSQL: новая методология разработки нереляционных баз данных / М. Фаулер, П. Дж. Садаладж ; пер. с англ. - М.: ООО Вильямс, 2016. – 192 с.

4. Verizon 2016 Data Breach Investigations Report Cybersecurity isn't just for security experts. The C-level guide to what you need to know. [Електронний ресурс] // – Режим доступу: http://www.verizonenterprise.com/resources/reports/rp_dbir-2016-executive-summary_xg_en.pdf (12.01.2017).

5. Глобальное исследование утечек информации в первом полугодии2016 года. [Електронний ресурс] // – Режим доступу: https://www.infowatch.ru/report2016_half (11.01.2017).

6. Global IT Security Risks Survey 2015: The current state of play [Електронний ресурс] // – Режим доступу: http://media.kaspersky.com/en/business-security/it-security-risks-survey-2015.pdf (11.01.2017).

7. Rohilla S., Mittal P.K. Database Security: Threads and Challenges. Intern. Journ. of Advanced Research in Computer Science and Software Engineering, 2013, vol. 3, iss. 5, pp. 810–813.

8. Burtescu E. Database security – attacks and control methods. Journ. of Applied Quantitative Methods, 2009, vol. 4, no. 4, pp. 449–454.

9. Баранчиков, А. И. Алгоритмы и модели ограничения доступа к записям баз данных / А. И. Баранчиков, П. А. Баранчиков, А. Н. Пылькин. - Москва : Горячая линия-Телеком, 2011. – 181 с.

10. Поляков А.М. Безопасность Oracle глазами аудитора: нападение и защита / А.М. Поляков. М.: ДМК Пресс, 2014. – 336 с.

11. Смирнов С.Н. Безопасность систем баз данных /С.Н. Смирнов. М.: Гелиос АРВ, 2007. – 352 с.

12. Полтавцева М.А. Безопасность баз данных: проблемы и перспективы / М.А. Полтавцева, А.Р. Хабаров // Программные продукты и системы. – 2016. – № 3.– С. 36-41.

13. Axon Tadd,Dillon Drew Understanding and Selecting a Tokenization Solution, Securosis, L.L.C., 2010. – 33pp.

14. Top Ten Database Security Threats. IMPREVA 2015 [Електронний ресурс] // – Режим доступу: http://www.imperva.com/docs/wp_topten_database_threats.pdf ( 16.01.2017).

15. Разоблачение мифа о безопасности NoSQL СУБД. Журнал «Хакер» [Електронний ресурс] // – Режим доступу: https://xakep.ru/2012/04/13/exposure-nosql-db( 12.01.2017).

Надійшла 20.02.2017 р. Рецензент: д.т.н., проф. Єрохін В.Ф.